19. (Not Amended) The computer-readable medium of claim 17, wherein a tag portion of the self-relative numeric reference indicates whether the first object and the second object have a same or different contiguity.

20. (Once Amended) The computer-readable medium of claim 19, wherein:

the tag portion includes bits of the tagged self-relative numeric reference that are less significant than bits used for an offset portion; and

the tag portion contains one of at least a first tag value indicating that the first object is contiguous and a second value indicating that the second object is non-contiguous, wherein a difference of the first value and the second value is congruent to 2^{N-1} modulo 2^{N}.

REMARKS

By this amendment, claims 1-20 are pending, in which claims 1, 3, 5, 11, 13-15, and 20 are amended. Care was exercised to avoid the introduction of new matter.

The Office Action mailed April 25, 2002 objected to the claim numbering under 37 CFR § 1.126 and rejected claims 5 and 15 under 35 U.S.C. § 102 as anticipated by *Brownell et al.* (US 6,009,266), claims 1-3, 6-9, 11-13, and 16-19 as obvious under 35 U.S.C. § 103 based on *Brownell et al.* in view of *Carter et al.* (US 5,845,331). The indication of allowable subject matter in claims 4, 10, 14, and 20 is noted with appreciation.

In response to the objection under 37 CFR § 1.126, claims 8-17 and 21-26 are renumbered as 5-14 and 15-20 respectively. Furthermore and unrelated to the merits of the prior art rejections, claims 14 and 20 have been amended to fix a typographical error in the exponent,

and claims 1 and 5 have been amended to provide a proper antecedent. No new matter is introduced.

The rejection of claims 1-4 and 11-14 is respectfully traversed because neither *Brownell et al.* nor *Carter et al.* teach or otherwise suggest the limitations of the claims. For example, independent claims 1 and 11 recite: "generating the first tagged machine pointer as a sum including the **tagged** numeric reference and the second tagged machine pointer" (emphasis added). The Office Action is correct to recognize that "Brownell does not teach generating a pointer as a sum of a numeric value and a second pointer." However, *Carter et al.* too fails to disclose this limitation because the offset used in the Load Effective Address (LEA) instruction disclosed in col. 7:23-34 is not "tagged" as recited in claims 1 and 11. This failing is evident in FIG. 2B, in which offset **48** is added only with the 54-bit address portion of pointer **42**. Furthermore, the non-address bits of pointer **42** are not summed with any value in FIG. 2B.

The rejection of claims 5 and 15, as amended, as anticipated by *Brownell et al.* is respectfully traversed. Claims 5 and 15 have been amended to define the claimed references as "numeric references that encode locations of referenced object as offsets from objects from objects that reference the referenced objects." No new matter is introduced because adequate descriptive support may be found throughout the specification, including p. 15, lines 12-13. By contrast, *Brownell et al.* in FIG. 5b and col. 13:49-65 discloses a persistent object reference **510** that includes an object kind field **511**, a host name **512**, a locator identification number **514**, an object key **516**, and a sub-object identifier **518**. None of these fields, however, "encode locations of referenced object as offsets from objects from objects that reference the referenced object" as presently recited in claims 5 and 15. Furthermore, since the object referenced by the persistent object reference **510** is stored on another computer as indicated by the host name **512** (see col.

13:59-65), it would not even be obvious to modify the *Brownell et al.* persistent object reference **510** to include such a recited offset between objects.

Furthermore, there is no motivation to modify *Brownell et al.*'s disclosure of persistent object references **510** to use the pointer arithmetic of the decidedly different and non-analogous guarded pointers described in *Carter et al.* as applied against claims 1-3, 6-9, 11-13, and 16-19. The guarded pointers described in *Carter et al.* are used in a "single virtual address space shared by all processes" (col. 4:62-63) of a single machine. By contrast, the *Brownell et al.* persistent object reference **510** is used to reference an object stored on another computer as indicated by the host name **512**. (col. 13:59-65) The *Carter et al.* pointer arithmetic with an offset does not make sense for any of the fields of the persistent object reference **510**, including the object kind field **511**, a host name **512**, a locator identification number **514**, an object key **516**, and a sub-object identifier **518**.

Dependent claims 2-4, 7-10, 12-14, and 17-20 are allowable for at least the same reasons as their independent claims and are separately patentable on their own merits.

For example, claims 3, 9, 13, and 19 recite a "**tag portion**" that "that indicates whether the first object and the second object have a same or a different contiguity." Although the Office Action cites sections of *Brownell et al.* purportedly for disclosure of same or different contiguity,[1] nothing in *Brownell et al.* discloses that this information is stored in a **tag portion** of a reference as required in the claims.

Therefore, the present application, as amended, overcomes the objections and rejections of record and is in condition for allowance. Favorable consideration is respectfully requested. If any unresolved issues remain, it is respectfully requested that the Examiner telephone the

[1] Though, in fact, merely residing in a same or different process is irrelevant to the notion of contiguity.

undersigned attorney at 703-425-8516 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

DITTHAVONG & CARLSON, P.C.

7/25/2002
Date

Stephen C. Carlson
Attorney/Agent for Applicant(s)
Reg. No. 39929

10507 Braddock Rd
Suite A
Fairfax, VA 22032
Tel. 703-425-8516
Fax. 703-425-8518

APPENDIX

1. (Once Amended) A method of generating a first tagged machine pointer to a first object referenced by a second object, said method comprising the computer-implemented steps of:

fetching a tagged numeric reference stored within the second object based on a second tagged machine pointer that points to the second object; and

generating the first tagged machine pointer as a sum including the tagged numeric [value] reference and the second tagged machine pointer.

2. (Not Amended) The method of claim 1, wherein the sum further includes a predetermined constant.

3. (Once Amended) The method of claim 1, wherein [the step of fetching a tagged numeric reference includes fetching] the tagged numeric reference [that] includes a tag portion that indicates whether the first object and the second object have a same or a different contiguity.

4. (Not Amended) The method of claim 3, wherein:

the tag portion includes bits of the first tagged numeric reference that are less significant than bits used for an offset portion; and

the tag portion contains one of at least a first tag value indicating that the first object is contiguous and a second value indicating that the second object is non-contiguous, wherein a difference of the first value and the second value is congruent to 2^{N-1} modulo 2^{N}.

5. (Once Amended) A method of managing memory, comprising the computer-implemented steps of:

storing a plurality of objects in a memory; and

storing references between the objects in the memory as [self-relative] numeric references that encodes locations of referenced objects as offsets from objects that reference the referenced objects.

6. (Not Amended) The method of claim 5, further comprising the step of calculating a pointer difference between a first machine pointer to a first object and a second machine pointer to a second object to produce a self-relative numeric reference.

7. (Not Amended) The method of claim 5, wherein the step of calculating a pointer difference between a first machine pointer to a first object and a second machine pointer to a second object to produce a self-relative numeric reference includes the step of calculating the pointer difference between a first tagged machine pointer to the first object and a second tagged machine pointer to the second object to produce a tagged self-relative numeric reference.

8. (Not Amended) The method of claim 7, wherein the pointer difference further includes a predetermined constant.

9. (Not Amended) The method of claim 7, wherein a tag portion of the self-relative numeric reference indicates whether the first object and the second object have a same or different contiguity.

10. (Not Amended) The method of claim 9, wherein:

the tag portion includes bits of the tagged self-relative numeric reference that are less significant than bits used for an offset portion; and

the tag portion contains one of at least a first tag value indicating that the first object is contiguous and a second value indicating that the second object is non-contiguous, wherein a difference of the first value and the second value is congruent to 2^{N-1} modulo 2^N.

11. (Once Amended) A computer-readable medium bearing instructions for generating a first tagged machine pointer to a first object referenced by a second object, said instructions arranged, when executed, to cause one or more processors to perform the steps of:

fetching a tagged numeric reference stored within the second object based on a second tagged machine pointer that points to the second object; and

generating the first tagged machine pointer as a sum including the tagged numeric [value] reference and the second tagged machine pointer.

12. (Not Amended) The computer-readable medium of claim 11, wherein the sum further includes a predetermined constant.

13. (Once Amended) The computer-readable medium of claim 11, wherein [the step of fetching a tagged numeric reference includes fetching] the tagged numeric reference [that] includes a tag portion that indicates whether the first object and the second object have a same or a different contiguity.

14. (Once Amended) The computer-readable medium of claim 13, wherein:

the tag portion includes bits of the first tagged numeric reference that are less significant than bits used for an offset portion; and

the tag portion contains one of at least a first tag value indicating that the first object is contiguous and a second value indicating that the second object is non-contiguous, wherein a difference of the first value and the second value is congruent to $[2^{N-14}]$ $\underline{2^{N-1}}$ modulo 2^N.

15. (Once Amended) A computer-readable medium bearing instructions for managing memory, said instructions arranged, when executed, to cause one or more processors to perform the steps of:

storing a plurality of objects in a memory; and

storing references between the objects in the memory as [self-relative] numeric references that encodes locations of referenced objects as offsets from objects that reference the referenced objects.

16. (Not Amended) The computer-readable medium of claim 15, said instructions further arranged to cause said one or more processors to perform the step of calculating a pointer difference between a first machine pointer to a first object and a second machine pointer to a second object to produce a self-relative numeric reference.

17. (Not Amended) The computer-readable medium of claim 15, wherein the step of calculating a pointer difference between a first machine pointer to a first object and a second machine pointer to a second object to produce a self-relative numeric reference includes the step

of calculating the pointer difference between a first tagged machine pointer to the first object and a second tagged machine pointer to the second object to produce a tagged self-relative numeric reference.

18. (Not Amended) The computer-readable medium of claim 17, wherein the pointer difference further includes a predetermined constant.

19. (Not Amended) The computer-readable medium of claim 17, wherein a tag portion of the self-relative numeric reference indicates whether the first object and the second object have a same or different contiguity.

20. (Once Amended) The computer-readable medium of claim 19, wherein:

the tag portion includes bits of the tagged self-relative numeric reference that are less significant than bits used for an offset portion; and

the tag portion contains one of at least a first tag value indicating that the first object is contiguous and a second value indicating that the second object is non-contiguous, wherein a difference of the first value and the second value is congruent to $[2^{N-14}]$ $\underline{2^{N-1}}$ modulo 2^N.